**Amendments to the Specification**

Please replace the paragraph at page 3, line 3, with the following rewritten paragraph.

Logon expiration is optional. However, most systems utilize some sort of session time-out tracking. This is particularly true in communications networks that employ communications protocols that do not inherently track session-state information. These networks are called "stateless". Most asynchronous communications networks are stateless.

a1

Please replace the paragraph at page 11, line 18, with the following rewritten paragraph.

The exemplary session-state manager implementation [[is ]]does not store a user's actual session-state information on any tier in a stateless network. Rather a Web server creates and delivers a one-way encrypted token to a user on a client of that server. Rather than including session-state information, the token incorporates a representation or a digest of the user's session-state information.

a2

Please replace the paragraph at page 13, line 17, with the following rewritten paragraph.

Scalability is a major advantage of a Web farm. As a site becomes more popular, additional Web servers can be added to the Web farm to support the additional load. A Web farm typically includes a Web database 250. These databases include central information that is shared by all of the Web servers.

a3

Please replace the paragraph at page 14, line 9, with the following rewritten paragraph.

a4

The exemplary embodiment of the session-state manager is[[ be]] integrated into the operation of a Web server. For example, the exemplary embodiment uses one or more COM (Component Object Model) components called from within dynamic pages such as ASP (Active Server Page).

Please replace the paragraph at page 17, line 12, with the following rewritten paragraph.

a5

The exemplary implementation of the session-state manager uses session-state tokens, rather than storing session-state information. These tokens are generated by a Web server and sent to a user. These tokens are subsequently received from the user and examined by the server.

Please replace the paragraph at page 18, line 13, with the following rewritten paragraph.

a6

Fig. 4 shows an incremental series of buckets at 400. In this exemplary series of buckets, each bucket is one hour long. Of course, the exact length of each bucket is an implementation detail that can be varied based upon the needs of each implementation. Assuming a fixed number of buckets before timeout occurs, the shorter buckets will lead to a shorter timeout period and the longer buckets will lead to a longer timeout period.

Please replace the paragraph at page 19, line 11, with the following rewritten paragraph.

a7

In the exemplary embodiment of the session-state manager uses time buckets as shown in Fig. 4 as part of the input to create tokens. This results in a situation where the token that is generated depends on the time bucket when the token is created. When the time bucket changes, a different token will be created. As explained in ~~create~~ greater detail below, this can be used to test for logon expiration.

Please replace the paragraph at page 21, line 6, with the following rewritten paragraph.

a8

At 512, the Web server gets the user's UserID that identifies the user of the client. This UserID may have been supplied by the user or [[is ]]may be retrieved from a database. The UserID may or may not be equivalent to the "username" used for logon authentication.

Please replace the paragraph at page 21, line 10, with the following rewritten paragraph.

a9

At 514, the Web server gets a code key (i.e., "secret string" or "trapdoor key"). This code key is defined data that will be used with the TimeID and the UserID so that it is more difficult to decode the encoded token and determine ~~and what~~ the TimeID and UserID. This code key may be statically or dynamically designated. If the code key is dynamically designated, it is preferable that code key be tracked carefully so that compared tokens are based upon the same code key.

Please replace the paragraph at page 22, line 6, with the following rewritten paragraph.

A10

Assuming that $T_{encrypted}$ is the encrypted token; *N[]* is a function [[the]]that takes a given number of bits; and *H[]* is a cryptographic hash function, the generation of the encrypted token of the exemplary embodiment may be represented by this formula:

Please replace the paragraph at page 24, line 1, with the following rewritten paragraph.

A11

One-way encryption schemes are those where the encrypted data cannot be decrypted. *Applied Cryptography* by Bruce Schneier (John Wiley & Sons, Inc., 1994) (p. 27) describes a one-way encryption scheme as one that is "relatively easy to compute but significantly harder to undo or reverse." It also says that[[ that]] "hard" means "it would take millions of years to compute...." In general, [[O]]one-way encryption schemes are far more secure than two-way encryption schemes.

Please replace the paragraph at page 24, line 7, with the following rewritten paragraph.

A12

Examples of one-way encryption schemes that may be used with the exemplary implementation of the session-state manager include a 128-bit MD5 hash, Secure Hash Algorithm (SHA), or any other cryptographically strong one-way hash function. The preferred one-way encryption scheme is fast and produces results that are apparently randomly distributed.

Please replace the paragraph at page 25, line 19, with the following rewritten paragraph.

Alternatively, the token may be unencrypted. In other words, the token may be plain text or plain data. However, this plain data may be encoded so that its meaning is not obvious absent additional information. For example, the encoded token may be a plain data reference to a look-up table.

Please replace the paragraph at page 26, line 7, with the following rewritten paragraph.

When the client makes a request, the client sends that token to the Web server. The data stored on the client is much smaller than with existing techniques that store actual session-state information on Tier A. In this exemplary embodiment, only about ten bytes of data are stored on the client.

Please replace the paragraph at page 28, line 1, with the following rewritten paragraph.

At 720, the Web server compares the new confirmation token with the received token. If they match, then a new token is issued and sent to the client at 722. Issuing a new token can mean: specifying the most-recently-generated token as the new token to be sent to the client; or generating a new token to be sent to the client. After that, the user is allowed access to the desired Web page or other resources at 724.

Please replace the paragraph at page 32, line 13, with the following rewritten paragraph.

a16

Issuing a new, user-associated TimeID can mean: specifying the most-recently-designated TimeID as the new user-associated TimeID to be sent to the client; or designating a new user-associated TimeID to be sent to the client. After that, the user is allowed access to the desired Web page or other resources at 824.

Please replace the paragraph at page 33, line 15, with the following rewritten paragraph.

a17

Again, this describes an alternative embodiment of the session-state manager. This alternative embodiment employs non-encrypted tokens that track[[s]] only logon expiration. This alternative embodiment does not necessarily have a high degree of security and it does not track user identification and logon validation.